TDS-88-06

Real-Time Intersection Simulation with Computer Graphics Displays

Ontario

Ministry of Transportation

Research and Development Branch

89-01222/

Technical Report Documentation Page

Real-Time Intersection Simulation with Computer Graphics Displays

Author:	IBI Group
Number:	TDS-88-06
Date of Publication:	November, 1988
Ministry Contact:	Alex Ugge (416) 235-4698
Abstract:	The feasibility of developing an interactive real-time intersection simulation and graphics display system was studied. Simulation would be used to generate traffic flow patterns and measurements of the network performance, and a graphical display used to convey information about the behaviour of the network in a meaningful, understandable way. Operating interactively and in real-time would allow the user to make changes in the parameters defining the network and immediately observe the results. Two levels of system sophistication, basic and extended, were identified and the required functions for each defined. The basic system is suitable for demonstration purposes or analyzing one or two intersections, while the extended system is required for a network of many intersections. Existing simulation/graphics programs were reviewed to determine their suitability in a real-time interactive system. Only two models were potentially suited to this application. Further investigation determined that both models lacked certain key features and were therefore not suitable. The requirements for developing an entirely new system were studied in terms of the simulation, graphics and control subsystems and their integration into a single package. It was concluded that a colour graphics/simulation system, developed from both new and existing subsystems, would be feasible. Hardware and software requirements were outlined and an implementation program for the basic system prepared. The development of a limited demonstration/evaluation system as an initial step was recommended to assess the utility of the model and provide guidelines for future expansion.
Comments:	Part of Research and Development Branch Project 25133.
Key words:	traffic simulation, computer graphics, intersections
Distribution:	MTO Head Office and Regions, key ministry officials, traffic management officials, researchers
Copyright Status:	Crown copyright © 1988 Ministry of Transportation

TDS-88-06

Real-Time Intersection Simulation with Computer Graphics Displays

IBI Group

Published by
The Research and Development Branch
Ministry of Transportation of Ontario
Hon. Ed Fulton, Minister
D.G. Hobbs, Deputy Minister

Published without prejudice as to the application of the findings.
Crown copyright reserved; however, this document may be reproduced for non-commercial purposes with attribution to the Ministry.

For additional copies, contact:
The Editor, Technical Publications
Room 320, Central Building
1201 Wilson Avenue
Downsview, Ontario
Canada M3M 1J8

Telephone: (416) 235-3480
CNCP Dialcom Electronic Mail: 23:MIN006

November 1988

Table of Contents

1.0 Introduction 1
1.1 Computer Technology Development 2
1.2 Feasibility Study Objectives 3

2.0 System Requirements 4
2.1 Functional Requirements 4
2.2 Hardware Environment 10
2.3 Software Environment 10

3.0 Review of Existing Intersection Simulation/Graphics Display Systems 12
3.1 Texas Simulation/Graphics System 13
3.1.1 Texas Microcomputer Requirements and Execution Time 15
3.1.2 Suitability of the Texas Model 15
3.2 GTRAF Simulation/Graphics System 19
3.2.1 GTRAF Hardware Requirements 20
3.2.2 GTRAF Model Suitability 20
3.3 Suitability of Existing Simulation/Graphics Programs 24

4.0 Development of a New Simulation/Graphics Display System 25
4.1 Simulation Models 25
4.1.1 Existing Microscopic Microcomputer Simulation Models 26
4.1.1.1 NETSIM Program 27
4.1.2 New Simulation Program Development 28
4.2 Graphics Displays 29
4.2.1 Graphics Processing 29
4.2.2 Creating the Intersection Network Display 30
4.2.3 Existing Graphics Subsystems 31
4.3 Internal and External Control Features 32
4.3.1 Simulation Program Control 32
4.3.2 External Control 32
4.3.3 Internal Control 33
4.4 System Configuration 33
4.4.1 Hardware Requirements 33
4.4.2 Software Requirements 35
4.4.3 Required NETSIM Modifications 35

5.0 Implementation Program 37
5.1 System Design and Development 38
5.1.1 Functional Specifications 38
5.1.2 Detailed Design 39
5.1.3 Development, Integration and Testing 40
5.1.4 Documentation 40
5.2 Implementation Timeframe and Budget for a Basic System 41
5.3 Immediate Actions 41

List of Exhibits

2.1 Summary of Simulation/Graphics Functions 5
2.2 Basic System 8
2.3 Extended System 9
3.1 Summary of Texas Simulation/Graphics System Features 16
3.2 Summary of GTRAF Simulation/Graphics System Features 21
4.1 System Configuration 34
5.1 Implementation Schedule for Basic System 43

REAL-TIME INTERSECTION SIMULATION WITH COMPUTER GRAPHICS DISPLAYS

1.0 INTRODUCTION

IBI Group was commissioned by the Ontario Ministry of Transportation to examine the feasibility of developing an interactive real-time analytical system capable of graphically displaying the operation of a network of intersections. In addition this system would generate measures of effectiveness to quantify the performance of an intersection network. Such information can assist a Traffic Engineer in understanding and improving intersection operations.

Because field measurements of an intersection network are difficult (if not impossible) to do, various mathematical models have been developed to approximate the actual situation. These models typically consist of computer programs that can be used to analyze existing networks or to predict the effects of changes in intersection geometry, traffic signal timings, vehicle technology changes, etc.

Most intersection network analysis programs use a simulation technique and process information in a "batch" mode. An initial data set is created which describes the properties of the road network, the rules governing vehicle behaviour, and the timing of the signals. This information is processed by the computer for a predetermined length of time and used to create an output file summarizing traffic flows, queue lengths, and various engineering measurements of network performance. Because of the method used to process the data, the Traffic Engineer cannot modify any of the operation parameters once the program is executed, nor can he observe the behavior of the network during the analysis.

1.1 COMPUTER TECHNOLOGY DEVELOPMENT

Until recently, intersection network analysis simulation programs have generally been designed to operate on mainframe computers such as the IBM 360/370. However, developments in low-cost, high-capacity microcomputers such as the IBM PC and AT has prompted the modification of existing mainframe simulation packages, as well as the development of entirely new packages designed for operation specifically on these and similar machines.

Output information from intersection simulation packages typically consist of pages of data which can be printed out and reviewed directly on a video screen. Although this type of output can provide the traffic engineer with all the required information to assess the performance of a particular network, it cannot actually illustrate the network nor can it show the behaviour of vehicles or other system components. Recent developments in microcomputer graphics displays have opened up the possibility of actually showing the network and its behaviour on a monochrome or colour video screen. Graphic subsystems are now available with sufficient resolution and colour capability to display the intersection network, traffic signals, and other items pertinent to the network. All that is typically required for such a graphics system is an IBM PC or AT and a high resolution colour monitor. Less expensive, lower resolution monitors can be used but the quality of the display will suffer significantly. Technology advances and market competitiveness are rapidly forcing the price of high resolution monitors down.

Coupling a good graphics display package with a sophisticated simulation package allows the traffic engineer to generate quantitative

information on the performance of the intersection network as well as allowing him to actually view its behaviour under simulated operating conditions. He can observe the effects of changes in signal timings, vehicle routings, and vehicle distributions without committing the resources necessary to implement these alternatives in the field. In addition, the traffic engineer has complete control over the experimental conditions and can change as many or as few aspects of the situation as he chooses without affecting the others.

1.2 FEASIBILITY STUDY OBJECTIVES

The objective of this feasibility study is to determine the functional requirements of an interactive real-time intersection simulation/display system and to prepare an implementation program for it. To meet these objectives, both necessary and desired operational and display functions are developed and compared with those found in existing simulation programs and graphics packages. Such information is used to determine whether or not these existing packages can be integrated together to form a complete real-time simulation/display system and, if not, what modifications are required to develop a complete package. After the system requirements have been determined a preliminary implementation program including scheduling and budgeting will be developed.

Chapter 2 of the report outlines the functional requirements of the simulation/graphics program and describes the hardware and software environments for the system. Chapter 3 reviews existing intersection simulation/graphics display systems in terms of applicability to the requirements outlined in Chapter 2. With a view to creating an entirely new package Chapter 4 reviews simulation subsystems, graphic subsystems and control interfaces, and outlines a system hardware configuration. Chapter 5 contains the implementation schedule.

2.0 SYSTEM REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

The functional requirements of the real-time intersection simulation graphics display system can be divided up into two levels: basic and extended. Exhibit 2.1 summarizes the operational features of each level in terms of interfacing with the user, traffic signal control, graphics capability, simulation parameters, and measures of effectiveness.

The basic system, illustrated in Exhibit 2.2, is suitable for a limited demonstration, and incorporates a microscopic real-time network simulation program to generate data for the display of operation of the intersection on a high resolution colour graphics display terminal. All control parameters (such as traffic signal cycles, offset, splits, etc.) can either be stored in the program or input from an external source (keyboard or other controller). The microscopic simulation program used in the system must contain enough operational parameters (vehicle dynamics, vehicle behaviour, fleet composition, network geometrics and traffic control) to provide realistic and accurate modelling of actual traffic behaviour. It should also allow the operator to zoom in on any desired portion of the intersection. The basic system would allow one or two intersections to be controlled and analyzed simultaneously and would graphically show all vehicles.

The extended system has the capability of simulating a whole network of intersections. It allows the user to create intersection features directly on the graphics display screen using some form of cursor control (keyboard, mouse, etc.). Signals from the detector loops are transmitted not only to the simulation program, but also to an external controller to provide input for its operation. This system would also have the option of displaying or not displaying the vehicles. Where the vehicles were not displayed, passage over a detector loop would be

EXHIBIT 2.1

SUMMARY OF SIMULATION/GRAPHICS FUNCTIONS

GRAPHIC SIMULATION	BASIC	EXTENDED
USER'S INTERFACE		
o user friendly (e.g. menu selection) for simulation/testing/data entry	0	
o full graphics capabilities	0	
o graphics editor to create intersection features		0
o detector sizing and placement		0
o store outputs to allow post processing and analysis	0	
o graphic display in real time	0	
o error checking	0	
CONTROL		
o simulation program	0	
o manual control (keyboard)	0	
o external controllers		
- 170's (with all options)	0	
- Nema	0	
o software interface to candidate control algorithms		0
GRAPHICS SYSTEM		
o high resolution	0	
o variable scale		0
o zoom capability	0	

SUMMARY OF SIMULATION/GRAPHICS FUNCTIONS

GRAPHIC SIMULATION	BASIC	EXTENDED
SIMULATION MODEL		
General:		
o real time processing and display	0	
o replicate simulation	0	
o define initial condition	0	
Traffic Flow Modeling:		
o lane changing	0	
o vehicle dynamics	0	
o fleet composition	0	
o gap acceptance	0	
o specify arrival patterns	0	
o specify volumes	0	
o volume composition	0	
o turning movements	0	
o transit	0	
Geometrics:		
o approach lanes	0	
o turning lanes	0	
o curb lane designations	0	
o approach grades	0	
o detector placements	0	
o mid-block sources/sinks	0	
o lane closing/obstructions	0	
o railway crossings	0	

SUMMARY OF SIMULATION/GRAPHICS FUNCTIONS

GRAPHIC SIMULATION	BASIC	EXTENDED
Geometrics (Cont'd):		
o fixed intersection geometrics	0	
o definable intersection geometrics	0	
o one intersection	0	
o two intersections	0	
o network of intersections		0
Control Features of Simulation:		
o coordination	0	
o pedestrian actuation	0	
o skipable phases	0	
o eight phase dual ring	0	
o overlap	0	
o preemption	0	
o ability to blank vehicles and show detector activations only		0
Measures of Effectiveness:		
o delay	0	
o speed	0	
o stops	0	
o queue lengths	0	
o energy consumption	0	
o pollution	0	
o travel time	0	
o user definable	0	

Exhibit 2.2 – Basic System

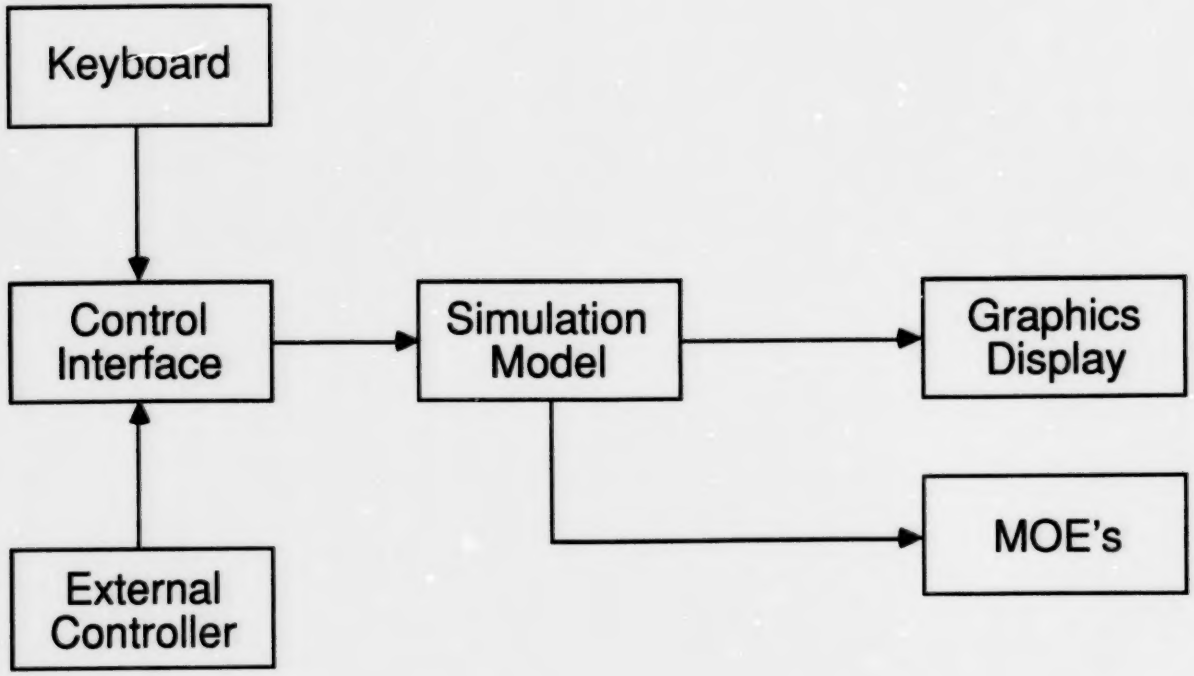
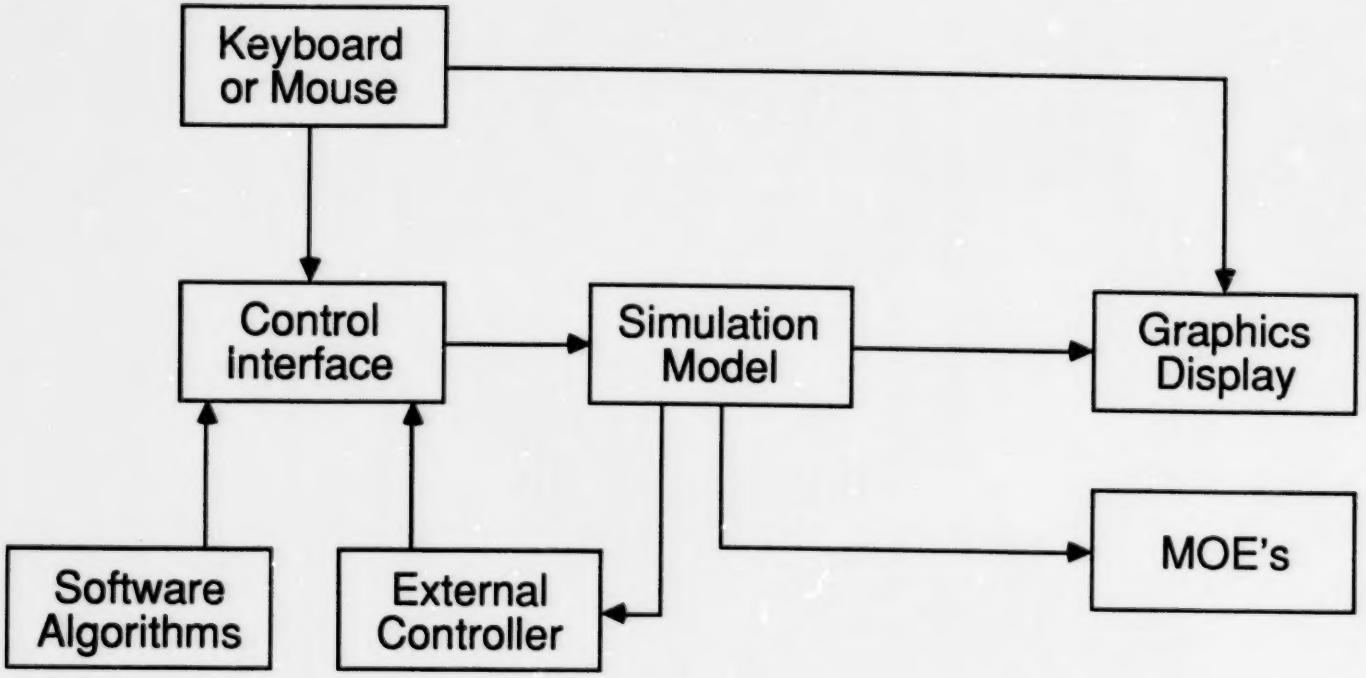


Exhibit 2.3 – Extended System



indicated by an on-screen "flash" of the loop. The extended system is illustrated in Exhibit 2.3.

2.2 HARDWARE ENVIRONMENT

The two major considerations governing the selection of a hardware environment for the real-time intersection/graphics display program are cost and accessibility (both in terms of obtaining the hardware components themselves and accessing the system). High speed mainframe computers with extensive data processing capabilities have typically been used for running complex simulation programs. Unfortunately, computing costs on a mainframe computer are relatively high (when compared to those on a microcomputer) and many traffic engineers, particularly those in the field, do not have easy access to such a machine.

Relatively low cost, high performance microcomputers are now available which can perform most of the functions of a mainframe computer, albeit at a somewhat slower speed. Microcomputer technology has advanced to the stage where complex simulation and graphics programs can be executed at speeds approaching those of a mainframe. Because of these advancements (and expected future ones), the simulation/graphics program should be designed to operate on a microcomputer to ensure maximum current and future flexibility and accessibility.

2.3 SOFTWARE ENVIRONMENT

The software environment should use low cost, easily available and commonly accepted programs. The operating system should be as close to an industry standard as possible, easily obtainable, and should be chosen for its compatibility or similarity to operating systems currently being used by the majority of traffic engineers. The software used in the simulation/graphics package should, wherever possible, use existing

programs which are standard or at least familiar to most Traffic Engineers. It should be nonproprietary and readily accessible through some form of distribution network (a good distribution network is essential for ensuring that all users receive updated versions of the software promptly).

3.0 REVIEW OF EXISTING INTERSECTION SIMULATION/GRAPHICS DISPLAY SYSTEMS

IBI Group contacted a number of different transportation authorities to determine what simulation/graphics packages were available worldwide as well as their capabilities and limitations. Agencies contacted included:

- the Transport and Road Research Laboratory of England's Department of Transport;
- the Australian Road Research Board;
- the Institute of Industrial Science at the University of Tokyo;
- the Federal Department of Traffic Analysis and Regulation (Department Analyse et Regulation du Trafic) of France;
- the United States Federal Highways Administration (FHWA);
- the transportation division of the Department of Civil Engineering of the University of Toronto;
- the Department of Civil and Mineral Engineering of the University of Minnesota;
- the University of Florida Transportation Research Centre;
- the Centre for Transportation Research, Bureau of Engineering Research of the University of Texas at Austin.

Replies received from contact persons within these agencies indicated that there were two microscopic intersection simulation/graphics systems available which could possibly meet the aforementioned functional requirements. The first of these is the TEXAS (Traffic, EXperimental, Analytical, Simulation) model developed at the Centre for Transportation Research, Bureau of Engineering Research of the University of Texas at Austin, and the second was the GTRAF system developed for the United States Federal Highway Administration.

3.1 TEXAS SIMULATION/GRAPHICS SYSTEM

The TEXAS microscopic intersection simulation program is a user friendly system incorporating a colour graphics feature. The original program was developed in 1977 for operation on a mainframe computer and did not incorporate a graphics feature. In 1986 a user-friendly version incorporating the colour graphics feature was developed for use on an IBM PC or AT compatible computer. A new version of this program (also designed for operation on IBM PC or AT compatible machines) is expected in late 1987, and will be distributed by the McTrans division of the University of Florida Transportation Research Center. The microcomputer version of TEXAS also incorporates an interactive data entry subprogram which allows the user to enter network parameters directly with a keyboard and video terminal.

The microcomputer version of TEXAS is subdivided into five main processing modules:

- GEOPRO (geometry) which defines the geometric intersection characteristics needed for simulation. The user specifies the lengths of inbound and outbound lanes on the approaches, the allowed directional traffic movements, and the orientation of the intersection legs. GEOPRO computes vehicle paths along the approaches and within the intersection;

- DVPRO (driver-vehicle) uses a uniform probability distribution to generate attributes for each individual driver-vehicle unit including driver class, vehicle class, desired outbound intersection leg, and lateral inbound lane position. A normal distribution is used to assign a desired speed to each individual driver-vehicle unit. Units are sequentially ordered by log-in time (the time of entry onto an inbound lane) as defined by the input of a selected headway distribution. The number of driver-vehicle units generated depends upon the input traffic volume and the time to be simulated;

- o SIMPRO (simulation) simulates the traffic behaviour of each unit according to the momentary surrounding conditions including applicable traffic control device indications. The entry velocity of each unit is set so that it will neither exceed a selected desired speed nor collide with the unit immediately ahead of it. A unit enters with its selected desired speed unless the vehicle immediately ahead of it is decelerating, in which case the entrance velocity is set to a value less than the desired one.

  Unit accelerations and decelerations are also controlled by SIMPRO as is unit response to traffic control devices. Traffic control devices incorporated into this subprogram include stop signs, pre-timed signals, and actuated signals.

  SIMPRO also computes the instantaneous location, speed, and acceleration of each vehicle, and generates statistics about delays and queue length to be used for evaluation of the performance of traffic at the intersection. This information is created at preset intervals and stored in an output file;

- o EMPRO (emissions) incorporates models to predict the instantaneous vehicle emissions of carbon monoxide, hydrocarbons, oxides of nitrogen, and fuel flow for both light-duty and heavy-duty vehicles. These computations are made by accessing vehicle behaviour information from SIMPRO;

- o an animated graphics display subsystem which allows the information generated by SIMPRO to be displayed graphically in real-time, or in stop action, on a screen driven by an IBM PC compatible computer. With this display the user can study the overall traffic performance at an intersection or examine in detail the behaviour of an individual vehicle in the traffic stream.

Output from the TEXAS model includes the instantaneous speed, location, and time relationship for every simulated vehicle. The data is written into an external file for use by the emissions processor or for other applications. The output information can also be displayed graphically in real-time, or in stop action, on a screen driven by an IBM PC or AT computer.

3.1.1 TEXAS Microcomputer Requirements and Execution Time

GEOPRO, DVPRO, SIMPRO, and EMPRO are written totally in ANSI standard FORTRAN 77. The animation and graphics processor was written in assembly language to enable a faster execution of the display. TEXAS is designed to operate on an IBM PC-XT or PC-AT with at least 512K of RAM, a math co-processor, a colour graphics adapter and colour graphics monitor, or an enhanced graphics adapter and monochrome monitor. A printer and microsoft DOS 3.1 are also required by the current configuration.

Execution time for normal intersection simulation run on an IBM PC-XT is slightly greater than the real-time that is simulated. Processing time on an IBM PC-AT is approximately 1/2 real-time.

3.1.2 Suitability of the TEXAS Model

The main features of the TEXAS model are summarized in Exhibit 3.1. Although the TEXAS model incorporates many of the system requirements, it has insufficient control and graphics capabilities. It cannot handle pedestrian movements and cannot generate or process detector actuation information.

The most serious limitation of the TEXAS model concerns its graphics capability. The graphics animation program is a post-processor designed to display the behaviour of the intersection network after the simulation has been completed rather than during its execution. In addition, the graphics subsystem does not have the resolution required for adequate display of a complex intersection network. Also, the graphics routines are written in assembler and are thus hardware dependent.

Because of these limitations the TEXAS model is not considered suitable as a true real-time intersection simulation with computer graphics display program.

EXHIBIT 3.1

SUMMARY OF TEXAS SIMULATION/GRAPHICS SYSTEM FEATURES

GRAPHIC SIMULATION	TEXAS
USER'S INTERFACE	
o user friendly (e.g. menu selection) for simulation/testing/data entry	0
o full graphics capabilities	0
o graphics editor to create intersection features	X
o detector sizing and placement	X
o store outputs to allow post processing and analysis	0
o graphic display in real time	0
o error checking	0
CONTROL	
o simulation program	0
o manual control (keyboard)	X
o external controllers	
- 170's (with all options)	X
- Nema	X
o software interface to candidate control algorithms	X
GRAPHICS SYSTEM	
o high resolution	X
o variable scale	X
o zoom capability	X

0 = Feature Supplied
X = Feature Not Supplied
? = Unknown

SUMMARY OF TEXAS SIMULATION/GRAPHICS SYSTEM FEATURES

GRAPHIC SIMULATION	TEXAS
SIMULATION MODEL	
General:	
o real time processing and display	X
o replicate simulation	0
o define initial condition	0
Traffic Flow Modeling:	
o lane changing	0
o vehicle dynamics	0
o fleet composition	0
o gap acceptance	0
o specify arrival patterns	X
o specify volumes	0
o volume composition	0
o turning movements	0
o transit	X
Geometrics:	
o approach lanes	0
o turning lanes	0
o curb lane designations	X
o approach grades	0
o detector placements	0
o mid-block sources/sinks	X
o lane closing/obstructions	X
o railway crossings	X

SUMMARY OF TEXAS SIMULATION/GRAPHICS SYSTEM FEATURES

GRAPHIC SIMULATION	TEXAS
Geometrics (Cont'd):	
o fixed intersection geometrics	0
o definable intersection geometrics	0
o one intersection	0
o two intersections	X
o network of intersections	X
Control Features of Simulation:	
o coordination	0
o pedestrian actuation	X
o skipable phases	?
o eight phase dual ring	?
o overlap	?
o preemption	?
o ability to blank vehicles and show detector activations only	X
Measures of Effectiveness:	
o delay	0
o speed	0
o stops	0
o queue lengths	0
o energy consumption	0
o pollution	0
o travel time	0
o user definable	X

3.2 GTRAF SIMULATION/GRAPHICS SYSTEM

GTRAF is an interactive computer graphics system designed to operate in conjunction with the readily available microscopic intersection simulation program called NETSIM (discussed in detail in Chapter 4). It allows the user to review the input data and results generated by the NETSIM model. The combination of GTRAF and NETSIM may be executed either on microcomputers having graphics capabilities or on any mainframe computer supporting FORTRAN and graphics. The software used in this system includes both FORTRAN 77 and the Graphics Kernel System (GKS), both of which have been adopted by the American National Standards Institute (ANSI).

The developers of GTRAF recognized that although NETSIM was probably the most widely used microscopic intersection simulation program in North America, it was not particularly "user-friendly". Because a typical input data set for NETSIM may take anywhere from several hours to several days to create and check, the developers of GTRAF have attempted to reduce the amount of checking time required by providing a far more meaningful graphical display of the input data. Part or all of the network can be displayed directly on the video screen as a link-node diagram. If desired, the user can obtain a detailed graphical display of a particular link showing the number of lanes, its length, and vehicle discharges.

Output from the NETSIM model typically consists of a number of reports (generated at pre-specified intervals) summarizing the behaviour of the system over the previous time period and providing a cumulative summary of a number of measures of network performance. Because of the complexity of these output files, it can be very difficult for the traffic engineer to gain a qualitative understanding of either the behaviour or performance of this system. GTRAF aids the user by displaying the network (again in a link-node format) and selected output results. It can also

create bar graphs or line graphs to illustrate important performance measurements such as link speeds, vehicle delay, vehicle movements, etc. Step-wise pseudo-animation of the output results can be obtained by running through the output files sequentially.

It should be stressed that NETSIM and GTRAF are separate programs. GTRAF is initially used to verify the input data, the NETSIM model is then run, and GTRAF is used once more to review the simulation results. GTRAF cannot be used to display the results of the NETSIM program during its execution.

3.2.1 GTRAF Hardware Requirements

GTRAF is designed to be run on an IBM PC-AT (or compatible) computer and is not generally transportable to other hardware configurations. The NETSIM model is designed to operate on an IBM PC or AT machine having at least 336K of memory, two disk drives, a printer, and a disk operating system of Version 2.0 or higher.

3.2.2 GTRAF Model Suitability

The main features of the GTRAF program are illustrated in Exhibit 3.2. With the exception of the external control features, GTRAF incorporates almost all of the system requirements. Unfortunately, like the TEXAS program, the graphics subsystem operates as a post processor and can only display the intersection network behaviour after simulation has been completed. GTRAF also does not have the capability of displaying the movements of vehicles throughout the system; a feature which is essential for meaningful representation of the behaviour of the network. Because of these limitations, GTRAF is not considered suitable for use as a real-time intersection simulation with computer graphics display program.

EXHIBIT 3.2

SUMMARY OF GTRAF SIMULATION/GRAPHICS SYSTEM FEATURES

GRAPHIC SIMULATION	GTRAF
USER'S INTERFACE	
o user friendly (e.g. menu selection) for simulation/testing/data entry	O
o full graphics capabilities	O
o graphics editor to create intersection features	O
o detector sizing and placement	X
o store outputs to allow post processing and analysis	O
o graphic display in real time	O
o error checking	O
CONTROL	
o simulation program	O
o manual control (keyboard)	X
o external controllers	
- 170's (with all options)	X
- Nema	X
o software interface to candidate control algorithms	X
GRAPHICS SYSTEM	
o high resolution	O
o variable scale	O
o zoom capability	O

O = Feature Supplied
X = Feature Not Supplied
? = Unknown

SUMMARY OF GTRAF SIMULATION/GRAPHICS SYSTEM FEATURES

GRAPHIC SIMULATION	GTRAF
SIMULATION MODEL	
General:	
o real time processing and display	X
o replicate simulation	X
o define initial condition	X
Traffic Flow Modeling:	
o lane changing	0
o vehicle dynamics	0
o fleet composition	0
o gap acceptance	0
o specify arrival patterns	0
o specify volumes	0
o volume composition	0
o turning movements	0
o transit	0
Geometrics:	
o approach lanes	0
o turning lanes	0
o curb lane designations	X
o approach grades	0
o detector placements	0
o mid-block sources/sinks	0
o lane closing/obstructions	X
o railway crossings	X

SUMMARY OF GTRAF SIMULATION/GRAPHICS SYSTEM FEATURES

GRAPHIC SIMULATION	GTRAF
Geometrics (Cont'd):	
o fixed intersection geometrics	0
o definable intersection geometrics	0
o one intersection	0
o two intersections	0
o network of intersections	
Control Features of Simulation:	
o coordination	0
o pedestrian actuation	X
o skipable phases	?
o eight phase dual ring	0
o overlap	?
o preemption	?
o ability to blank vehicles and show detector activations only	X
Measures of Effectiveness:	
o delay	0
o speed	0
o stops	0
o queue lengths	0
o energy consumption	0
o pollution	0
o travel time	0
o user definable	X

3.3. SUITABILITY OF EXISTING SIMULATION/GRAPHICS PROGRAMS

None of the simulation/graphics models reviewed are suitable for a real-time intersection simulation and graphics display system. The two best models, TEXAS and GTRAF, have many of the desired features but cannot display the behaviour of the intersection network during simulation execution (both programs use a post processor to generate the graphics displays).

Researchers at the University of Pittsburgh and Clarkson Institute of Technology are currently working on developing two new simulation/graphics models. However, contact with persons at both of these institutions has indicated these models will initially be simulation only (graphics may be added later). Work on the simulation portion of both these models is expected to be completed by mid-1988.

4.0 DEVELOPMENT OF A NEW SIMULATION/GRAPHICS DISPLAY SYSTEM

Because there are no existing programs available satisfying the requirements for the real-time simulation and graphics system, it is evident that a new model is required to provide the features of the basic and/or extended systems. The model itself can be broken down into four main subcomponents which include:

- o the simulation software;
- o the graphics display;
- o control and interface features; and
- o integration of the above into a single package.

These subcomponents are discussed in greater detail in this chapter.

4.1 SIMULATION MODELS

There are three types of simulation models commonly used to analyze road traffic networks. The first of these is the macroscopic model which treats the traffic stream as a continuum and conceptualizes it as a fluid. In such a model individual vehicles are not identified and elements of the network may even contain fractional vehicles. Macroscopic models perform the simulation by relating relatively simple mathematical and statistical formulae. Because these types of simulations are simple, they are very economical in terms of computer storage and very fast in execution. Unfortunately, they cannot represent the traffic stream in the detail desired by most traffic engineers.

Platoon models are the next step up in simulating the actual situation. In these models vehicles are grouped into platoons, and the program keeps track of the location, speed, acceleration, and size of the

platoon. Because the platoon speed is usually a function only of the general density of vehicles in its vicinity, complicated car-following calculations are avoided. Platoon models provide a good balance between accurate representation of the actual system and minimization of computing requirements.

The most detailed type of simulation is the microscopic model. Every vehicle in the network is identified and its position, speed, acceleration and other characteristics are stored. This information is used in conjunction with geometrical information about the system to provide a simulation which provides the level of accuracy and detail required by traffic engineers. Unfortunately, these types of simulations require a considerable amount of input data as well as computer processing time.

For the purposes of integration into a real-time intersection analysis and display system, the only type of simulation that can meet the functional requirements is the microscopic model. The other two cannot model a complex intersection with the degree of detail and accuracy required by a traffic engineer. It was noted previously that microscopic simulations require a considerable amount of computer processing time. With the advent of low cost, high speed personal computers processing time is no longer a major concern.

4.1.1 Existing Microscopic Microcomputer Simulation Models

A number of existing microscopic intersection simulation programs were reviewed to determine their suitability for incorporation into a real-time intersection simulation/graphics program. These programs were evaluated in terms of the features supplied, speed of execution, computing requirements, availability, and documentation. The majority of the programs, such as VETRAS, VPT, and SIGNET are designed to operate on mainframe computers. Some, like VETRAS, are event-oriented programs and

do not track the movement of individual vehicles throughout the system. The only available program which met all the requirements is NETSIM, the simulation program currently used for GTRAF.

4.1.1.1 NETSIM Program

NETSIM is a Fortran-coded software simulation program which describes the operational performance of vehicles travelling over a network of streets. It is available for a nominal charge from the McTrans Center at the University of Florida Transportation Research Center. The program is fully microscopic and updates the status of the system in one second (simulation time) intervals. The current version is designed for use on any IBM PC or AT compatible computer.

The program incorporates all of the traffic flow modelling, geometric, and measure of effectiveness features required for the minimum system. However, only internal control is supplied (although it may be possible to add external control by modifying the program).

The current version of NETSIM was released in May 1986 and was written in relatively unstructured FORTRAN. An updated, structured version is expected in the near future (to be distributed through McTrans).

There are three main steps in the program execution:

- load and check the input data which describes the network, the type of vehicles, speeds, signal information, etc.;
- emit vehicles from source nodes until a prespecified traffic equilibrium condition is met;
- perform the simulation for the specified length of time. The simulation has seven processing sub-steps which iterate over the one second time period.

During the execution of the program, output files are generated at preset intervals detailing the behaviour of vehicles over the previous interval. A second type of report which summarizes cumulative statistical information is generated and stored at the end of a different preset interval. If desired, a number of specialized output options can be created for use in connection with particular applications of the model.

Benchmark tests of the NETSIM program were performed on a variety of microcomputers to evaluate its execution time. A 15 minute simulation of a two intersection network filled with vehicles was run on an IBM PC and on an AT compatible machine using an INTEL 80286 microprocessor. Actual execution time for the 15 minutes of simulation time on the IBM PC was 21.3 minutes and on the AT compatible was 4.7 minutes. Actual execution time on the IBM PC was somewhat longer than the simulation time for the sample intersection network. For the AT style machine execution time was significantly less than the simulation time (approximately 30%).

Because the simulation/graphics program is designed to operate in real-time, the simulation subsystem must operate at a speed faster than real-time. This condition becomes especially important if the same microprocessor is used to generate the graphics display and process input/output commands from external controllers.

The primary problem with NETSIM is that it requires a large amount of input data. Depending on the complexity of the intersection network being analyzed, it may take several hours or even several days to accumulate, verify, and input the required information.

4.1.2 New Simulation Program Development

An alternate solution to developing a simulation subsystem which operates at faster than real-time and incorporates all the features of the minimum, desired, or ultimate system is to generate a completely new

program. Although this program can be written with any conventional high level language (Fortran, Assembler, C, etc.), there exists a language called the General Purpose Simulation System (GPSS) designed especially to aid in the modelling of queuing systems. Creation of the simulation subsystem using this language would require extensive programming and debugging. Because the simulation subsystem must model some traffic phenomena not based on queuing theory, it may not even be possible to create an adequate program using GPSS. The work required to create a simulation subsystem using this language is likely to require too much budget to be considered feasible.

4.2 GRAPHICS DISPLAYS

With the advent of true 16 and 32 bit microcomputers, real-time graphics technology for personal computer based systems has evolved at a rapid pace. State-of-the-art technology allows performance of personal computer based systems to come close to the traditional minicomputer based workstation. The newest generation of colour graphics processors and subsystems designed for operation on microcomputer based systems are quickly approaching performance levels of the more powerful dedicated graphics workstations.

4.2.1 Graphics Processing

For the purpose of displaying real-time traffic flows on a typical intersection network, the microcomputer based graphics subsystem must have the capability of processing information generated by the simulation model and converting it to a graphical representation of streets, traffic signals, vehicles, etc. This requires very high speed graphics processing capability which, for a microcomputer, can only be obtained by using one of the latest generation of microprocessors.

Typical examples of state-of-the-art graphics microprocessors include the TEXAS Instruments TI-34010, Hitachi HD63484 and the INTEL 82786. A graphics processing subsystem built around one of these microprocessors would require at least two megabytes of random access memory for image capture and buffering. High speed direct memory access to the host bus is also essential so that vehicle and equipment status can be processed quickly. A minimum of 8 bytes per display pixel would be required in order to obtain realistic images.

As a minimum the colour monitor should have the capability of displaying 256 colours and have a resolution of 1024 X 768 to provide a suitable definition of vehicles and intersection control devices. It should work on a non-interlace 60Hz format (to avoid screen flickering) and should have a dot pitch less than 0.31. An optional requirement could be to integrate NTSC video input and output to allow standard images from a camera or other video medium to be displayed on the monitor. This type of monitor is typically referred to as a "very high resolution monitor".

Standard monitors, such as those used for IBM's Video Graphics Array (VGA) and Professional Graphics Adapter (PGA) systems do not have sufficient resolution.

4.2.2 Creating The Intersection Network Display

The background display (streets, signals, detector loops, and other static elements) can be created either with an interpreter program or by manually drawing the network on the screen. The interpreter program would use the pertinent geometry data stored in the simulation input file to create the intersection network on the monitor. This type of program can be expected to be complex and will require a considerable amount of programming and debugging.

Drawing the intersection network on the monitor using a digitizing tablet and/or mouse device would require considerably less software. With this type of system it is important to ensure that the intersection network drawn on the monitor corresponds with the one stored in the simulation program. A serious disadvantage of using the digitizing tablet/mouse combination is that a new portion of the network must be drawn every time there is a geometry change. The required tablet resolution would be between 200 and 1,000 dots per inch. The mouse technology could be optical, electromechanical, or optimechanical.

4.2.3 Existing Graphics Subsystems

An investigation into existing graphics packages using the latest generation of microprocessors was conducted to determine if there was a commercially available package capable of meeting the graphics requirements of the system. It was concluded that despite the fact extensive sophisticated graphics hardware was available, an adequate software package was not. To solve this problem IBI Group has developed a high resolution colour graphics generation subsystem which can be integrated into any PC or AT based microcomputer workstation. This program can display both static and moving elements with any combination of 256 colours. One of the requirements of the graphics subsystem is that it have the capability of moving elements in real-time. As more elements are added to the system processing time increases and the apparent movement of those elements decreases. The program currently being used by IBI Group can handle up to approximately 25 different moving elements on a second by second basis before processing time becomes greater than real-time. This may not be suitable for intersections having a demand greater than the 25 vehicle limit. If the dynamic elements (vehicles) were updated every 2 seconds, their number could be increased to approximately 50. Alternatively, graphics processing time can be reduced by writing the required routines in assembler; an option that would increase development costs.

4.3 INTERNAL AND EXTERNAL CONTROL FEATURES

A major functional requirement of the real-time intersection simulation/graphics package is that it incorporate traffic signal control. Control can be from an external source (e.g. user or traffic controller) or from an internal source (test algorithms, preset signal timings, or detector loop data). The actual amount of control depends upon the chosen system (basic or extended).

In the basic system the only form of traffic signal control required are the preset timings stored in the simulation subsystem. In the extended system a greater level of internal and external control, not found in existing simulation and graphics programs, is needed. To provide for this additional level of control a separate software subsystem is required to provide an interface between the simulation/graphics program and control units.

4.3.1 Simulation Program Control

The simulation program should have the capability of storing and executing pre-set signal timing plans. This allows the program to operate automatically without input from either a keyboard or an external controller.

4.3.2 External Control

Three types of external control have been identified which include user, model 170 and NEMA. The user control would allow the traffic engineer to alter the signal timings from a keyboard during the execution of this simulation and observe the affects on traffic flow. The two levels of external control (170 and NEMA) have been chosen to represent actual traffic signal controllers used in the field. These controllers would typically be separate from the simulation/graphics

program but would be connected to it with a combination hardware/ software interface. Input for the external control units would come from volume and speed detector loop information generated by the simulation program.

4.3.3 Internal Control

In the extended system test algorithms designed to optimize signal timings during simulation execution would be incorporated. These algorithms would receive detector loop actuation information and use this data to optimize traffic flow patterns by altering signal timings as appropriate. Different algorithms would be used to represent various signal timing response plans.

4.4 SYSTEM CONFIGURATION

A typical configuration for the real-time graphics/simulation system is illustrated in Exhibit 4.1. The blocks shown in solid lines are required for all three levels of system complexity whereas the ones shown in dashed lines are only required for the basic and extended configurations.

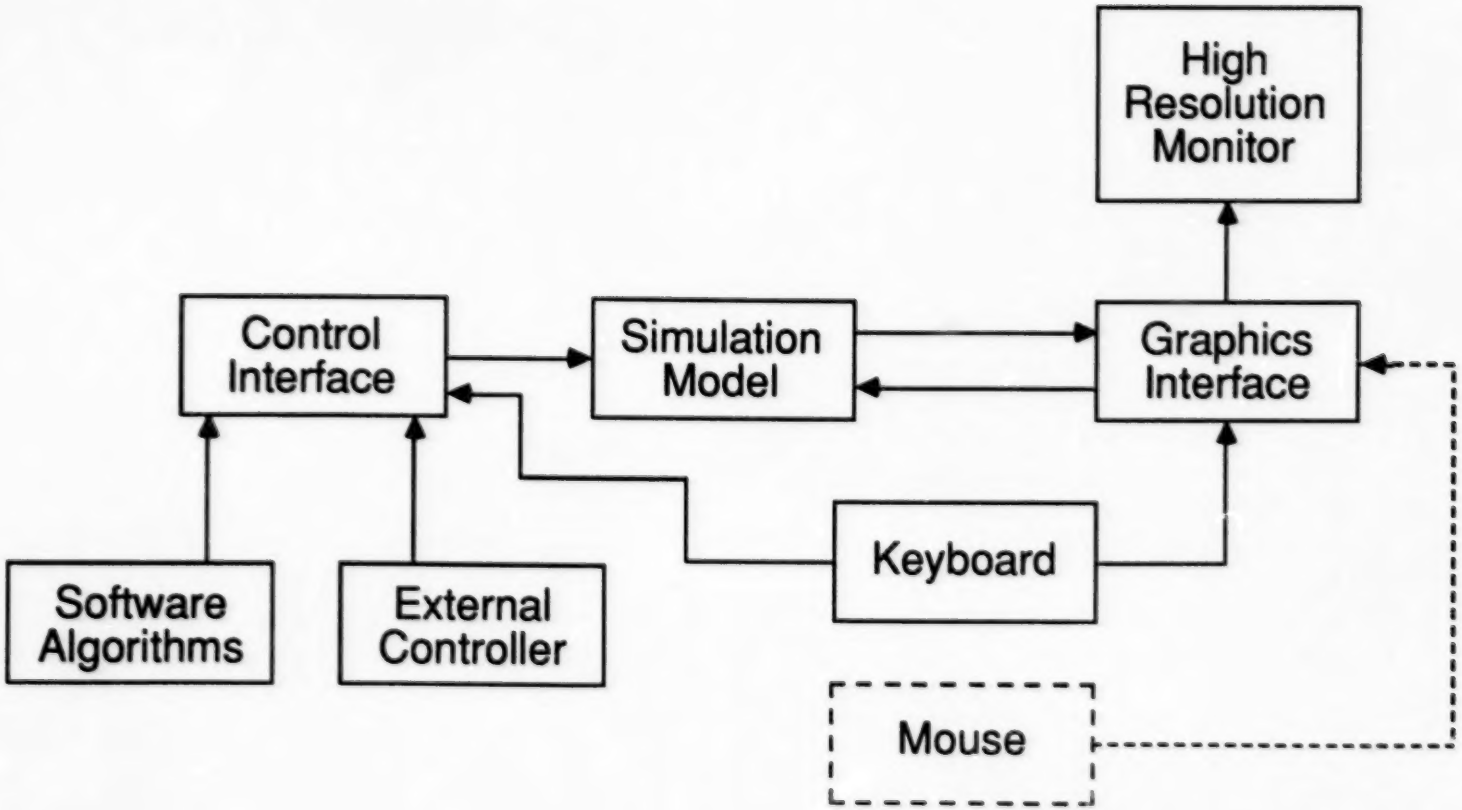
4.4.1 Hardware Requirements

The hardware requirements for the simulation/graphics system are as follows:

- an IBM PC-AT compatible microcomputer using either the Intel 80286 or 80386 microprocessor to run the simulation program and control interfaces. Because the main microcomputer must perform other functions besides simulation processing, it is essential it have multi-tasking capability and be able to run the simulation program faster than real-time;

Exhibit 4.1 – System Configuration

Note: Dashed lines indicate optional features.



- o a graphics processing subsystem built around the latest state-of-the-art graphics microprocessors (such as the Intel 82786 or similar) is required to provide the needed processing capability and speed;

- o a very high resolution (1024X768) colour monitor working on a non-interlace format to provide the required display resolution;

- o other interface hardware to provide communication between the main microcomputer, graphics processor, and external control units.

4.4.2 Software Requirements

The software requirements for the simulation/graphics system are as follows:

- o a readily available, commonly accepted simulation program having the features outlined in Section 2.1. Because of its many features and familiarity to most traffic engineers the NETSIM program appears to be the best one available for use as a simulation subsystem. Some modifications to this program will be required and outlined in Section 4.4.3;

- o a program written for the graphics microprocessor subsystem to provide the required static and moving displays. This program would also contain the interface to the simulation software;

- o a program to interface the simulation subsystem to external controllers (where required);

- o a main program to control overall operation of the simulation/graphics system.

4.4.3 Required NETSIM Modifications

NETSIM operates in an iterative manner and tracks the movements of all vehicles throughout the system. To integrate this program into the overall simulation/graphics system requires interrupting the simulation at

certain key points during each iteration. During these interruptions updated vehicle locations would be transmitted to the graphics subsystem and used to update the positions of the vehicles on the network. Where external control is desired detector loop actuation information would be extracted from NETSIM and transmitted to the controller. Internal signal timing module would be bypassed by this controller. Preliminary investigations into the NETSIM model have indicated that:

- o detector actuation information can be obtained from the DTCTR array which is called up in the DETECT and POLL subroutines (these subroutines record actuations of counter type and presence type detectors respectively);
- o external controller signal phase change commands can be inserted into the SIGT array;
- o the LINK array can provide data for graphical display of the links;
- o the V array contains the location of vehicles relative to upstream nodes. This information, when coupled with an interpreter program, can provide vehicle location information for the graphics display.

It is important to note that a new microcomputer version of NETSIM is planned for release in the near future. If the development of the graphics system were to proceed, it would be appropriate to build on this new version which is understood to be more structured and better suited to modification and integration into a larger graphics system.

5.0 IMPLEMENTATION PROGRAM

The objective of this project was to assess the feasibility of developing a real-time intersection simulation/display system, to determine the general requirements of such a system and to, if feasible, develop an implementation program. Based on the project findings, it can be concluded that:

- a colour graphics/simulation system would be feasible;
- such a system should make full use of new technology and readily available simulation tools.

The justification for proceeding with the development of such a system is dependent upon a number of items, including expected benefits and costs. The expected benefits of such a system might include traffic flow improvements, traffic engineering productivity enhancements, energy savings, accident benefits, etc. However, these benefits may be difficult to quantify. As is often the case in research, one cannot even fully quantify the potential benefits until a concept has been translated to a product and has been tested, revised and enhanced in the marketplace.

As an initial step, a limited demonstration/evaluation system would be developed to help quantify the potential costs and benefits of both the basic and extended systems. Such a system would include the same simulation model as the basic system, but would have fewer control features and be less expensive to develop. Signal timing plans would be stored in files and would be processed in a "batch" mode by the computer. External control would not be provided.

The main advantage of the demonstration/evaluation system is that it can provide feedback on the usefulness, applicability and viability of a real-time simulation/graphics system. In addition, it would stimulate interest for the continued development of a real-time simulation/graphics

system and may help identify desirable refinements and enhancements. Development of either the basic or extended systems would be accomplished by expanding the operation of the demonstration/evaluation system.

The costs of such a system are more identifiable and are comprised of:

- o system design and development costs;
- o hardware costs;
- o on-going maintenance and support.

To identify the design and development costs, the following section describes a typical systems development approach.

5.1 SYSTEM DESIGN AND DEVELOPMENT

A project of this scale and complexity requires the discipline of a proper systems development format which would include the following major tasks:

- o prepare functional specifications;
- o prepare detailed design;
- o develop, integrate and test the system;
- o prepare documentation.

5.1.1 Functional Specifications

The functional specifications are required to translate general system requirements into a more structured definition that is to be developed. The functional specification would include:

- o a description of the system features with respect to the menu format, general users interface, input requirements, reporting requirements, graphic screens, file contents, etc.;

- o the definition of the interfaces between the:

  - graphics and simulation;
  - the simulation and the controller;
  - the simulation and the user;

- o a review of the new NETSIM program to more properly understand its structure and the manner in which it could be integrated into the overall system;

- o a description of the overall systems management requirements including system security, backup, default values, etc.;

- o a description of the hardware environment taking into account the most recent technology;

- o preparation of a functional report and participation in selected meetings through the course of the project.

Throughout the preparation of the functional specifications, it is recommended that there be close contact with a designated users group. This users group would be particularly important in defining and refining the system functionality so as to ensure that the final product would be an effective tool.

At the completion of the functional specification task, there would be a formal sign-off on the design document. Also, the design, development and testing budget would be refined based on the results of the functional specifications.

5.1.2 Detailed Design

In this task, the hardware would be acquired and installed. Also, prototyping would be initiated in a variety of the major system modules include:

- o menus;
- o input screens;
- o reports;
- o graphics displays, etc.

Detailed design and testing would be initiated in each of the various major modules including the NETSIM model, graphics, the simulation graphics interface and the control and simulation interface. At the completion of this task, a further document would be prepared describing the detailed design which again would be used as a basis for signing-off on subsequent development. Also, the development, testing and integrating budget would be revised.

5.1.3 Development, Integration and Testing

In this task, all of the major modules would be integrated and tested. These would include the simulation model, the controller interface, the graphics module and interface, the users interface, etc.

5.1.4 Documentation

Documentation would be prepared and would include:

- o a users manual to assist potential users in defining input parameters, altering control strategies, interpreting the results, etc.;
- o a systems manual which would contain all source codes properly commented with appropriate structure charts, etc.

5.2 IMPLEMENTATION TIMEFRAME AND BUDGET FOR A BASIC SYSTEM

Exhibit 5.1 summarizes the estimated project schedule and budget for the basic system (based on the NETSIM model). As is apparent, the estimated timeframe for the project would be approximately 22 months including a demonstration and evaluation task. If the latter task were not included, the overall timeframe would be approximately 18 months.

The budget for the project is estimated at approximately $205,000 ± 10%, thereby providing a budget range for development to between $180,000 and $225,000. In addition, there would be hardware costs which would be in the range of $15,000 to $20,000, bringing the overall project cost to between $200,000 and $250,000.

5.3 IMMEDIATE ACTIONS

In light of the estimated timeframe and budget, it would seem appropriate that a number of possibilities be pursued prior to embarking on such an extensive development initiative. These should include:

- a survey of traffic engineers to determine if a graphics simulation product would be of value. This might be best conducted through the professional representative bodies such as ITE, supplemented by direct contact with operating divisions within selected jurisdictions such as Metro, Ottawa, etc.;

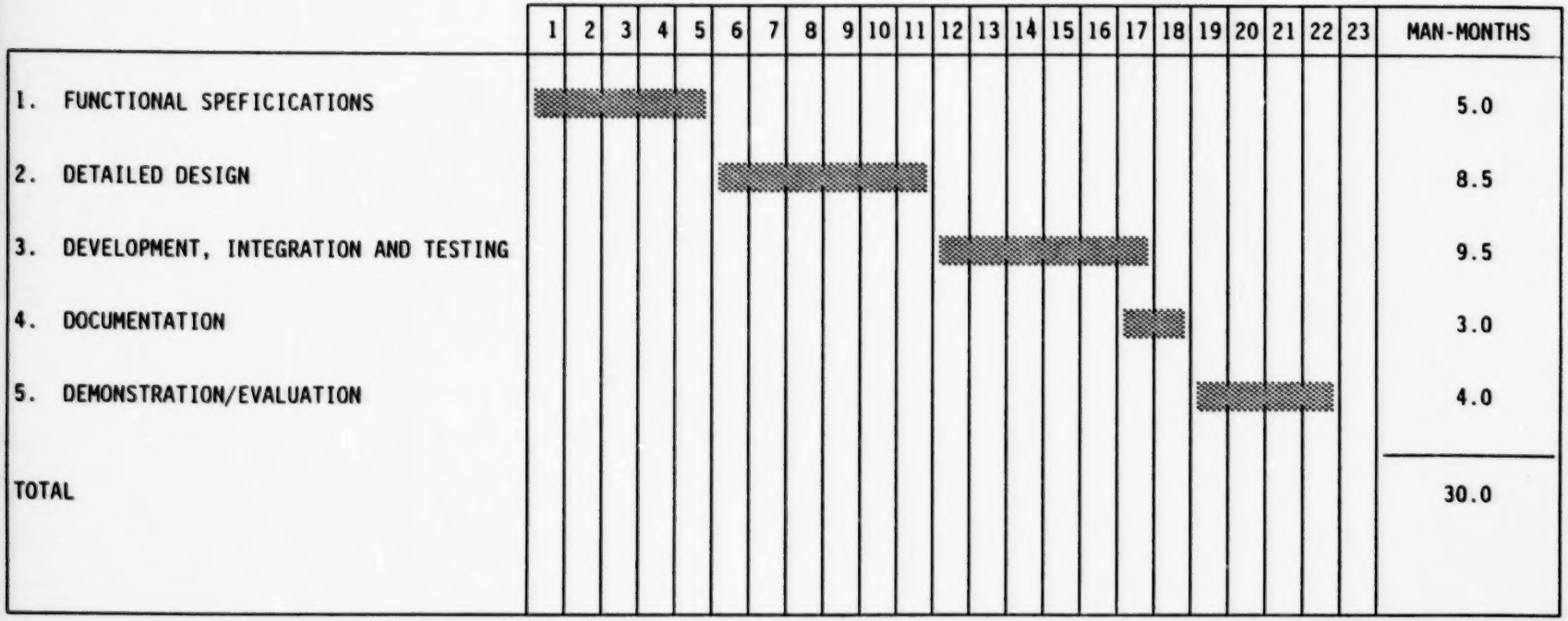
- because of the research nature of the initiative as well as the expected budget, it would be preferable if the project could be jointly sponsored with other agencies. The approaches that were made to selected agencies such as the FHWA were not very successful. This is unfortunate because it would provide synergy and also reduce the overall costs. Failing this, an option would be to pursue the joint development of the system with Transport Canada and/or Roads and Transportation Association of Canada;

- o incorporate the new NETSIM simulation program. As such, it may be most appropriate to await any decision to proceed until this program is available and the source code can be reviewed and the associated system development efforts refined.

Also, consideration might be given to the development of a portion of the desired system initially. Once this portion was proven, it could then be enhanced and other modules added in an incremental manner.

EXHIBIT 5.1

IMPLEMENTATION SCHEDULE FOR BASIC SYSTEM*



* Based on Proposed Application of NETSIM.